Elementary algebra related to the SAT problem

Samuel Lundqvist

Abstract

This paper deals with elementary algebraic properties of ideals in $\mathbb{Z}_2[x_1, \ldots, x_n]$ that contains the field polynomials $x_i^2 + x_i$ for $i = 1, \ldots, n$ without introducing the concept of Gröbner bases.

1 Introduction

In recent years, various authors have studied Gröbner bases as an alternative method to decide whether a boolean formula is satisfiable or not [1, 2, 3, 4, 8, 10, 15, 18]. The Buchberger algorithm, which is the most used algorithm to compute a Gröbner basis, takes as input a set of polynomials in a polynomial ring over a field $\Bbbk$ and returns a Gröbner basis for the ideal defined by the input polynomials. By using the Stone transformation, every logical formula can be interpreted as a set of polynomials in $\mathbb{Z}_2[x_1, \ldots, x_n]$, and it is easily proven that the original formula is satisfiable if and only if the unit 1 is not a member of the computed Gröbner basis.

However, the solvers based upon Gröbner bases are outperformed by the solvers which are based upon the DPLL algorithm [9]. The common way to explain this is by arguing that the DPLL-influenced algorithms has been much more studied, and that a lot of tricks have been invented to make the implementation effective, and that the Gröbner based methods needs time to be properly tuned. The author of this paper does not agree with this explanation and the answer is due to a basic property of Gröbner bases. A Gröbner basis does not only answer the question whether or not the corresponding boolean formula is satisfiable or not, it also gives the number of such assignments[1], that is, it answers the problem #SAT (also known as Sharp-SAT). Thus, as soon as the number of satisfiable assignments exceeds one, the Gröbner basis method reveals more information about the problem than we originally asked for.

This paper deals with various aspects of the one-to-one correspondence between the elements in the boolean ring $\mathbb{Z}_2[x_1, \ldots, x_n]/(x_1^2 + x_1, \ldots, x_n^2 + x_n)$ and the subsets of $\mathbb{Z}_2^n$. It turns out that the structure in this ring is so simple that much that can be done without using the theory of Gröbner bases.

As is well known, see for instance [1], ideals in $\mathbb{Z}_2[x_1, \ldots, x_n]$ such that $x_i^2 + x_i \in I$ for $i = 1, \ldots, n$, can be written in the form $I = (x_1^2 + x_i, \ldots, x_n^2 + x_n, f)$ for a polynomial $f \in \mathbb{Z}_2[x_1, \ldots, x_n]$. Using the Stone transformation, it follows that there is a one-to-one correspondence between the (truth tables of) boolean formulas in n variables and polynomials $f = m_1 + \cdots + m_s \in \mathbb{Z}_2[x_1, \ldots, x_n]$, where the exponent vector of each monomial m_i consists of zeroes and ones only.

[1] The number of valid assignments is equal to the number of monomials which are not divisible by the leading monomials of the elements in the Gröbner basis. Detecting this set of monomials is neglectable compared to the computation of the Gröbner basis.

We give elementary proofs, without using the Nullstellensatz, and explicit formulas of the one-to-one correspondence between ideals in $\mathbb{Z}_2[x_1,\ldots,x_n]$ polynomials in $\mathbb{Z}_2[x_1,\ldots,x_n]$ and the power set of $\mathbb{Z}_2^n$. Explicit formulas for the latter correspondence does not seem to have been examined before in the literature.

We give a family of normal form methods without using the notion of Gröbner bases and monomial orderings. This approach is totally different from the so called Gröbner Free approach in [3], see Remark 3.14.

We also give simple formulas for the quotients, sums, products and intersections of boolean ideals.

In the end of the paper we discuss computational aspects and compare our direct approach to the approach using Gröbner bases. The results are striking and in several examples, our naive three line long implementation in Macaulay2 outperforms both the special package written for boolean ideals in Macaulay2 and the PolyBoRi framework [2]. In some examples, the direct method is more than 10000 faster compared to the methods which relies on Gröbner basis theory.

2 Preliminaries

A boolean polynomial is a finite sum of boolean monomials. A boolean monomial is an element of the form $x_{i_1}\cdots x_{i_m}$, where $1 \leq i_1 < \cdots < i_m \leq n$, i.e. the coefficient vector consists of zeroes and ones only.

We evaluate a boolean polynomial f at affine points in $\mathbb{Z}_2^n$ by the notation $f(p)$ and we introduce the concept of zero set for a boolean polynomial which we denote by $V(f)$. In this case, we only consider elements in Z_2^n; hence

$$V(f) = \{p \in \mathbb{Z}_2^n \mid f(p) = 0\}.$$

We also use the notation $V(I)$, where I is an ideal in $\mathbb{Z}_2[x_1,\ldots,x_n]$. In this case $V(I)$ is, as usual, the common zero set of all elements in the ideal. Here we do not make the assumption that $V(I) \subseteq \mathbb{Z}_2^n$, but all ideals we consider will contain the field polynomials $x_i^2+x_i$ for $i = 1,\ldots,n$, so in practice $V(I)$ will actually be a subset of $\mathbb{Z}_2^n$. (Recall that if $I = (f_1,\ldots,f_m)$ is any ideal in $\mathbb{Z}_p[x_1,\ldots,x_n]$, then $V(x_1^p+x_1,\ldots,x_n^p+x_n,f_1,\ldots,f_m) = V(I)\cap\mathbb{Z}_p^n$.)

We say that two polynomials $f,g \in \mathbb{Z}_2[x_1,\ldots,x_n]$ are equal as boolean polynomials if $f \equiv g$ modulo $(x_1^2+x_1,\ldots,x_n^2+x_n)$. This is expressed as "$f = g$ as boolean polynomials".

If Ω is a subset of $\mathbb{Z}_2^n$, we denote by Ω^c the complement of Ω in $\mathbb{Z}_2^n$, and given an element $f \in \mathbb{Z}_2[x_1,\ldots,x_n]$ such that $f = m_1+\cdots+m_r$ as boolean polynomials, where each m_i is a boolean mononomial and $m_i \neq m_j$ for $i \neq j$, let $\mathrm{Suppb}(f) = \{m_1,\ldots,m_r\}$.

3 Boolean polynomials, boolean ideals and subsets of $\mathbb{Z}_2^n$

Here we state the main results of this paper. In Section 3.1 we give new and short proofs of known results needed in for the theory we develop in the remaining sections. Section 3.2 deals with algebraic aspects of boolean ideals. We also give a way to compute the normal form with respect to an ideal and we also give formulas for the common ideal

operations. In Section 3.3, we give some basic results related to the variety of a boolean ideal.

3.1 One-to-one correspondences

The aim of this section is to establish a one-to-one correspondence between subsets of $\mathbb{Z}_2^n$, boolean polynomials, and boolean ideals.

We begin by constructing a map Pol from the set of subsets of $\mathbb{Z}_2^n$ to the set of boolean polynomials.

$$\mathrm{Pol}(\Omega) = \begin{cases} 1 & \text{if } \Omega = \{\}, \\ 1 + \prod_{i=1}^{n}(x_i + p_i + 1) & \text{if } \Omega = \{p\}, \\ \mathrm{Pol}(\{v_1\}) \cdots \mathrm{Pol}(\{v_k\}) & \text{if } \Omega = \{v_1, \ldots, v_k\} \text{ and } k > 1. \end{cases}$$

Lemma 3.1. *Let Ω be a subset of $\mathbb{Z}_2^n$. Then*

$$V(\mathrm{Pol}(\Omega)) = \Omega.$$

Proof. When Ω is the empty set, we get $V(\mathrm{Pol}(\Omega)) = V(1) = \{\}$.

Suppose now that Ω consists of a single point p. We need to assure that $(\mathrm{Pol}(\{p\}))(p) = 0$ and that $(\mathrm{Pol}(p))(q) = 1$ when $q \neq p$. In the first case we get

$$(\mathrm{Pol}(p))(p) = 1 + \prod_{i=1}^{n}(p_i + 1 + p_i) = 1 + 1 = 0.$$

When $q \neq p$, let i be such that $q_i \neq p_i$. Then $p_i + 1 + q_i = 0$ and hence $(\mathrm{Pol}(p))(q) = 1$.

Suppose finally that Ω consists of $k > 1$ points; $\Omega = \{v_1, \ldots, v_k\}$. Again we need to assure that $\mathrm{Pol}(v_1) \cdots \mathrm{Pol}(v_k)(v_i) = 0$ for $i = 1, \ldots, k$ and that $\mathrm{Pol}(v_1) \cdots \mathrm{Pol}(v_k)(q) = 1$ for $q \in \mathbb{Z}_2^n \setminus \{v_1, \ldots, v_k\}$. But this follows since $\mathrm{Pol}(\{v_i\})(v_i) = 0$ and $\mathrm{Pol}(\{v_1\})(q) = \cdots = \mathrm{Pol}(\{v_k\})(q) = 1$ by above.

□

Theorem 3.2. *The maps V and* Pol *are inverses and establishes a one-to-one correspondence between subsets of $\mathbb{Z}_n^2$ and boolean polynomials.*

Proof. The cardinality of the number of subsets of $\mathbb{Z}_2^n$ equals 2^{2^n}. But this is the same number as the number of boolean polynomials in n variables. It follows by Lemma 3.1 that V and Pol are inverses. □

We will now prove the one-to-one correspondence between boolean polynomials and boolean ideals. The next lemma should be well known but is proven for completeness.

Lemma 3.3. *Let f be a polynomial in $\mathbb{Z}_2[x_1, \ldots, x_n]$. Then $f^2 = f$ as boolean polynomials.*

Proof. When f is a monomial or $f = 0$, it is clear that $f^2 = f$. Suppose instead that $|\,\mathrm{Suppb}(f)| = s > 1$, so that $f = m_1 + \cdots + m_s$. We get

$$f^2 = m_1^2 + \cdots + m_s^2 + 2\sum_{i \neq j} m_i m_j = m_1 + \cdots + m_s = f$$

as boolean polynomials. □

Theorem 3.4. *There is a one-to-one correspondence between boolean polynomials and boolean ideals.*

Proof. Let $I = (x_1^2 + x_1, \ldots, x_n^2 + x_n, f_1, \ldots, f_m)$. Let $f = (f_1 + 1) \cdots (f_m + 1) + 1$ and let $J = (x_1^2 + x_1, \ldots, x_n^2 + x_n, f)$. We aim to show that $I = J$.

To show that $I \subseteq J$ it suffices to show that $f_i \in J$ for all i. We have $f_i f = 0 + f_i$ as boolean polynomials since $f_i(f_i + 1) = 0$ by Lemma 3.3. But $f_i f \in J$, thus $f_i \in J$ and it follows that $I \subseteq J$.

To prove that $J \subseteq I$, we need to show that $f \in I$. But this follows from the identity

$$f = f_1(f_2 + 1) \cdots (f_m + 1) + f_2(f_3 + 1) \cdots (f_m + 1) + \cdots + f_{m-1}(f_m + 1) + f_m.$$

□

Given an boolean ideal $I = (x_1^2 + x_1, \ldots, x_n^2 + x_n, f)$, we call f *the defining polynomial* with respect to I.

The following is an immediate corollary.

Corollary 3.5. *Let $f_1 = 0, \ldots, f_m = 0$ be a system of equations, where each $f_i \in \mathbb{Z}_2[x_1, \ldots, x_n]$. The solution set in $\mathbb{Z}_2^n$ is equal to the solutions to the single equation $(f_1 + 1) \cdots (f_m + 1) + 1 = 0$ in $\mathbb{Z}_2^n$.*

If we combine Theorem 3.2 and Theorem 3.4, we get

Corollary 3.6. *There is a one-to-one correspondence between subsets of $\mathbb{Z}_2^n$ and boolean ideals.*

Remark 3.7. *This result should be compared to the one-to-one correspondence between radical ideals and algebraic varieties (over algebraically closed fields) which is a well known corollary from the Nullstellensatz.*

Remark 3.8. *As mentioned in the introduction, the results in this section are not new; Theorem 3.2 corresponds to Corollary 2.1.7 in [1] and Theorem 3.4 corresponds to Theorem 2.2.7 in [1]. However, our proof of Theorem 3.4 does not make use of the Nullstellensatz or the theory of Gröbner bases.*

3.2 Normal forms, ideal membership and other ideal operations

Computations within ideals in polynomial rings are almost always carried out using the strong and rich theory of Gröbner bases. In this section we show that it is possible to perform direct computations in boolean rings in order to compute normal forms, decide ideal membership, and perform the common ideal operations.

Lemma 3.9. $V(f) \cap V(g) = V(f + g + fg)$

Proof. Suppose that $p \in V(g)^c$. Then $(f+g+fg)(p) = 1$. Suppose instead that $p \in V(g)$. Then $(f + g + fg)(p) = f$. Hence $p \in V(f + g + fg)$ if and only if $p \in V(f) \cap V(g)$. □

Lemma 3.10. *Let f and g be polynomials in $\mathbb{Z}_2[x_1, \ldots, x_n]$. Then $V(f) \cap V(g) = \{\}$ if and only if $(f + 1)(g + 1) = 0$ as boolean polynomials.*

Proof. By Lemma 3.9, $V(f) \cap V(g) = V((f+1)(g+1)+1)$. By Theorem 3.2, $\mathrm{Pol}(\Omega) = 1$ if and only if $\Omega = \{\}$. Hence $1 = (f + 1)(g + 1) + 1$ as boolean polynomials, which is equivalent to the statement in the lemma. □

Lemma 3.11. *Let f and g be boolean polynomials. Then*

$$V(f+g) = (V(f) \cap V(g)) \cup (V(f)^c \cap V(g)^c).$$

Proof. If $p \in V(f) \cap V(g)$, then $(f+g)(p) = f(p) + g(p) = 0$, so $p \in V(f+g)$. If $p \in V(f)$ but $p \notin V(g)$ or vice versa, then $(f+g)(p) = 1$, so $p \notin V(f+g)$. Finally, if $p \in V(f)^c \cap V(g)^c$, then $(f+g)(p) = 1+1 = 0$, so $p \in V(f+g)$. □

Lemma 3.12. *Let f be a boolean polynomial. Then*

$$V(f+1) = V(f)^c.$$

Proof. Use Lemma 3.11 with $g = 1$.

□

Recall that a normal form of an element in a quotient ring $\Bbbk[x_1, \ldots, x_n]/I$ is a unique representative (in S) for the equivalance class in S/I containing the element. Once a Gröbner basis is computed for I, the normal form of the equivalance class containing the element f can be computed using the concept of reduction.

We now give a family of normal form methods without using the concept of Gröbner bases or monomial orderings.

Theorem 3.13. *Let I be a boolean ideal, i.e. $I = (x_1^2 + x_1, \ldots, x_n^2 + x_n, f)$, where f is a boolean polynomial. Let g be a polynomial in $\mathbb{Z}_2[x_1, \ldots, x_n]$. Then, for a fixed boolean polynomial h, the expression $(f+1)g + h$ is a normal form of g with respect to I and is denoted by $\mathrm{NF}_h(g, f)$.*

Proof. We need to show that if $g_1 + I = g_2 + I$, then $\mathrm{NF}(g_1, f) = \mathrm{NF}(g_2, f)$, or

$$(f+1)g_1 + h = (f+1)g_2 + h,$$

which is equivalent to

$$(g_1 + g_2)(f+1) = 0.$$

If $g_1 + I = g_2 + I$, then $g_1 + g_2 \in I$ and hence $(g_1 + g_2)(p) = 0$ if $p \in V(f)$. On the other hand, if $p \in V(f)^c$, then $(f+1)(p) = 0$. Thus $(g_1 + g_2)(f+1)$ is identically zero on $\mathbb{Z}_2^n$ and must be equal to zero by Theorem 3.2.

□

Remark 3.14. *The family of normal forms described in Theorem 3.13 are different from that in [3]. The major difference is that the method in [3] is computed with information about the variety of the ideal given a priori. The normal form is then computed using interpolation methods, and is expressed as a linear combination of the monomials outside the initial ideal with respect to some monomial order $\prec$. Similar methods, based upon the Buchberger-Möller algorithm [6] and the Cerlienco-Mureddu correspondence [7] are discussed in [14], where among others, an $O(nm + m^2)$ algorithm is given to compute separators of the input points. Here m denotes the number of points and n denotes the number of variables. (To compute the normal form of an element with respect to a set of separators is equivalent to evaluate the element at the input points.)*

Theorem 3.15. *Let I be a boolean ideal. Let f be the defining boolean polynomial of I. Let g be a polynomial in $\mathbb{Z}_2[x_1, \ldots, x_n]$ and let $J = (x_1^2 + x_1, \ldots, x_n^2 + x_n, g)$. The following are equivalent.*

1. $g \in I$.
2. $J \subseteq I$.
3. $V(I) \subseteq V(J)$.
4. $V(f) \subseteq V(g)$.
5. $f \cdot h = g$ *as boolean polynomials, for some polynomial* h.
6. $f \cdot g = g$ *as boolean polynomials.*
7. $V(f) \cap V(g+1) = \{\}$.
8. $(f+1) \cdot g = 0$ *as boolean polynomials.*
9. $\mathrm{NF}_0(g, f) = 0$ *as boolean polynomials.*
10. $\mathrm{NF}_h(g, f) = h$ *as boolean polynomials.*

Proof.
(1) $\Leftrightarrow$ (2): If $g \in I$, then $(x_1^2+x_1, \ldots, x_n^2+x_n, g) \subseteq I$. If $(x_1^2+x_1, \ldots, x_n^2+x_n, g) \subseteq I$, then especially $g \in I$.
(3) $\Leftrightarrow$ (4): Follows from Theorem 3.4 and Corollary 3.6.
(4) $\Rightarrow$ (2): It holds that $g(p) = 0$ for all $p \in V(f)$. Hence $g \in I$.
(2) $\Rightarrow$ (4): If $g \in I$ we have $g(p) = 0$ for all $p \in V(I)$. Hence $V(f) \subseteq V(g)$.
(4) $\Rightarrow$ (7): If $V(f) \subseteq V(g)$, then $V(f) \cap V(g)^c = \{\}$. But $V(g)^c = V(g+1)$ by Lemma 3.12.
(7) $\Leftrightarrow$ (8) : By Lemma 3.9, $V(f) \cap V(g+1) = \{\}$ is equivalent to $V((f+1)g+1) = \{\}$. By Theorem 3.4 this is equivalent to $(f+1)g + 1 = 1$.
(8) $\Leftrightarrow$ (9) : By definition.
(9) $\Leftrightarrow$ (10) : Add h on both sides in (10).
(6) $\Leftrightarrow$ (8): Add g on both sides in (6).
(6) $\Rightarrow$ (5): Obvious.
(5) $\Rightarrow$ (4): Since $V(f \cdot h) = V(f) \cup V(h)$ it follows that $V(f) \subseteq V(g)$.

□

We end this section by listing some results on operations on boolean ideals.

Theorem 3.16. *Let I and J be boolean ideals and let f and g be the defining polynomials of I and J respectively. Then*

1. $I : J = (x_1^2 + x_1, \ldots, x_n^2 + x_n, 1 + g + fg)$.
2. $I + J = ((x_1^2 + x_1, \ldots, x_n^2 + x, (f+1)(g+1) + 1)$.
3. $IJ = I \cap J = (x_1^2 + x_1, \ldots, x_n^2 + x, fg)$.

Proof.

1. If a is an element such that $ab \in I$ for all $b \in J$, then $V(f) \subseteq V(a) \cup V(b)$ for all $b \in J$, especially for $b = g$. But $V(f) \subseteq V(a) \cup V(g)$ if and only if $V(g)^c \cap V(f) \subseteq V(a)$. Hence $a \in (x_1^2 + x_1, \ldots, x_n^2 + x_n, 1 + g + fg)$ by Lemma 3.9.
   On the other hand, if a is an element in $(x_1^2 + x_1, \ldots, x_n^2 + x_n, 1 + g + fg)$, then $a \cdot b \in I$ for all $b \in J$ since $V(f) \subseteq (V(g)^c \cap V(f)) \cup V(g)$.

2. If p is a point in $V(I+J)$, then $p \in V(f) \cap V(g)$ or $p \in V(f)^c \cap V(g)^c$. But this happens if and only if $p \in (f+1)(g+1)+1$, hence $I+J = (x_1^2+x_1, \ldots, x_n^2+x_n, (f+1)(g+1)+1)$.

3. It holds trivially that $V(I \cap J) = V(I) \cup V(J)$ and $V(I) \cup V(J) = V(fg)$. Suppose that $a \in IJ$. Then $a = b_1c_1 + \cdots + b_mc_m$, where $b_i \in I$ and $c_i \in J$. Thus $a(p) = 0$ if $p \in V(I) \cup V(J)$, so $IJ \subseteq I \cap J = (x_1^2+x_1, \ldots, x_n^2+x_n, fg)$. On the other hand, it holds that $fg \in IJ$, hence $I \cap J \subseteq IJ$. This concludes the proof. Remark: It is a general result from commutative algebra that $\sqrt{IJ} = \sqrt{I \cap J}$.

□

Remark 3.17. *Notice that the defining polynomial of $I : J$ is equal to* $\mathrm{NF}_1(g, f)$.

Example 3.18. Let

$$I = (x_1^2+x_1, x_2^2+x_2, x_3^2+x_3, x_1x_2+x_3, x_1x_3+x_2, x_3+1)$$

and let

$$f = (x_1x_2+x_3+1)(x_1x_3+x_2+1)(x_2x_3+1+1)+1 = x_1x_2x_3+x_3+1,$$

so

$$I = (x_1^2+x_1, x_2^2+x_2, x_3^2+x_3, x_1x_2x_3+x_3+1).$$

Since $x_3 \in \mathrm{Suppb}(f)$, $V(f) \neq \{\}$ by Theorem 3.2. Let $g = x_1x_3+1$. Then $\mathrm{NF}_0(g, f) = (x_1x_2x_3+x_3)(x_1x_3+1) = x_1x_3+x_3 \neq 0$, hence $g \notin I$. However, the element $h = 1+x_3 \in I$, since $(x_1x_2x_3+x_3)(1+x_3) = 0$. Thus $V(x_1x_2x_3+x_3+1) \subseteq V(x_3+1)$.

In fact, $V(f) = \{(1,0,1),(0,0,1),(0,1,1)\}$, $V(g) = \{(1,0,1),(1,1,1)\}$ and $V(h) = \{(0,0,1),(0,1,1),(1,0,1),(1,1,1)\}$, and since $V(f) \cup V(g) = V(h)$, we obtain the factorization $fg = h$.

Let $J = (x_1^2+x_1, x_2^2+x_2, x_3^2+x_3, h)$. Thus $J \subseteq I$ and $V(I) \subseteq V(J)$. We get $I : J = (x_1^2+x_1, x_2^2+x_2, x_3^2+x_3, 1+h+fh) = (x_1^2+x_1, x_2^2+x_2, x_3^2+x_3, 1) = (1)$, which is expected, since the geometric interpretation of the colon operation is $V(I) \setminus V(J)$.

Indeed, $J : I = (x_1^2+x_1, x_2^2+x_2, x_3^2+x_3, 1+f+fh) = (x_1^2+x_1, x_2^2+x_2, x_3^2+x_3, 1+x_1x_2x_3)$ and $V(J : I) = V(J) \setminus V(I) = \{(1,1,1)\}$.

Remark 3.19. *If I is a boolean ideal, then there are no irreducible elements in the quotient ring $\mathbb{Z}_2[x_1, \ldots, x_n]/I$, since $[f] = [f]^2$ by Lemma 3.3. However, the elements* $[\mathrm{Pol}(\{v_i\})]$ *for $i = 1, \ldots, 2^n$ are special; every element $[f] \in \mathbb{Z}_2[x_1, \ldots, x_n]/I$ can be written as a product of elements of the form* $[\mathrm{Pol}(\{v_i\})]$ *and such a factorization is unique up to powers of the factors, indeed*

$$[f] = \prod_{v \in V(f)} [\mathrm{Pol}(\{v\})].$$

In general, the number of factorizations of a boolean polynomial f with $|V(f)| = m$ is huge. If we restrict ourself to only consider factorizations of the form $f = f_1 \cdots f_s$ (as boolean polynomials) where $V(f_i) \cap V(f_j) = \{\}$ for all $i \neq j$, then there are B_m such factorisations, where B_m denotes the mth Bell number.

Finally, the ideals $([\mathrm{Pol}(\{v_i\})])$ *are the only maximal ideals in $\mathbb{Z}_2[x_1, \ldots, x_n]/I$ and the remaining ideals are all non-prime.*

3.3 Computing the zero set of a boolean ideal/polynomial

We give some elementary results on how to get information about the variety from the structure of the ideal.

Lemma 3.20.

$$V((x_{i_1} - c_{i_1}) \cdots (x_{i_m} - c_{i_m})) = \{(p_1, \ldots, p_n) \,|\, p_{i_j} = c_{i_j} \text{ for at least one } j\}$$

$$|V((x_{i_1} - c_{i_1}) \cdots (x_{i_m} - c_{i_m}))| = 2^n - 2^{n-m}.$$

$$V((x_{i_1} - c_{i_1}) \cdots (x_{i_m} - c_{i_m}))^c = \{(p_1, \ldots, p_n) \,|\, p_{i_j} = c_{i_j} + 1 \text{ for all } j\}$$

$$|V((x_{i_1} - c_{i_1}) \cdots (x_{i_m} - c_{i_m}))^c| = 2^{n-m}.$$

Proof. The first part follows from the evaluation

$$[(x_{i_1} - c_{i_1}) \cdots (x_{i_m} - c_{i_m})](p_1, \ldots, p_n) = (p_{i_1} - c_{i_1}) \cdots (p_{i_m} - c_{i_m}).$$

The remaining parts are direct consequences of the first part. □

Lemma 3.11 and Lemma 3.20 give us the following (non-effective) recursive algorithm to compute the variety of a boolean polynomial.

Proposition 3.21. *Let f be a boolean polynomial and write $f = m_1 + \cdots + m_k$, where the $m_i's$ are monomials. Then*

$$V(f) = (V(m_1) \cap V(m_2 + \cdots + m_k)) \cup (V(m_1)^c \cap V(m_2 + \cdots + m_k)^c).$$

We will now give a result on the number of elements in the variety.

Lemma 3.22. *Let f be a boolean polynomial in n variables and let $p = (p_1, \ldots, p_n)$ be a point in $\mathbb{Z}_2^n$. Let $g = (x_1 + p_1 + 1) \cdots (x_n + p_n + 1)$. Then*

$$V(f + g) = \begin{cases} V(f) \setminus \{p\} & \text{if } p \in V(f), \\ V(f) \cup \{p\} & \text{otherwise.} \end{cases}$$

Proof. By Lemma 3.20, $V(g) = \mathbb{Z}_2^n \setminus \{p\}$ and $V(g)^c = \{p\}$. Hence

- $V(f) \cap V(g) = V(f)$ if $p \notin V(f)$ and $V(f) \cap V(g) = V(f) \setminus \{p\}$ otherwise.
- $V(f)^c \cap V(g)^c = \{p\}$ if $p \notin V(f)$ and $V(f)^c \cap V(g)^c = \{\,\}$ otherwise.

The proof now follows from Lemma 3.11. □

Lemma 3.23. *If $|V(f)| = 1$, then the monomial $x_1 \cdots x_n$ belongs to* $\mathrm{Suppb}(f)$.

Proof. A direct consequence of the second part of Lemma 3.1. □

Proposition 3.24. *Let f be a boolean polynomial. Then $|V(f)|$ is odd if and only if $x_1 \cdots x_n \in \mathrm{Suppb}(f)$.*

Proof. Let f be a boolean polynomial. Suppose that $x_1 \cdots x_n \notin f$. We aim to prove that $|V(f)|$ is even. If $f = 0$ we are done, so suppose that $f \neq 0$ and let p be a point in $V(f)$. By Lemma 3.22,

$$|V(f + (x_1 + p_1 + 1) \cdots (x_n + p_n + 1))| = |V(f)| - 1.$$

Clearly $x_1 \cdots x_n \in f + (x_1 + p_1 + 1) \cdots (x_n + p_n + 1)$. Thus $f + (x_1 + p_1 + 1) \cdots (x_n + p_n + 1) \neq 0$ and by Theorem 3.2, $V(f)$ is nonempty. Let q be a point in $V(f + (x_1 + p_1 + 1) \cdots (x_n + p_n + 1))$ and consider the polynomial

$$g = f + (x_1 + p_1 + 1) \cdots (x_n + p_n + 1) + (x_1 + q_1 + 1) \cdots (x_n + q_n + 1)).$$

By Lemma 3.22 it holds that $|V(g)| = |V(f)| - 2$. It also holds that $x_1 \cdots x_n \notin \mathrm{Suppb}(g)$. If $|V(g)|$ would be odd, then we could repeat this process until we reach an element h such that $|V(h)| = 1$ and $x_1 \cdots x_n \notin h$. But this would contradict Lemma 3.23. Hence $|V(f)|$ contains an even number of elements.

Suppose instead that $x_1 \cdots x_n \in f$. With a similiar process as the one above, the assumption that $|V(f)|$ is even would imply that $x_1 \cdots x_n \in \mathrm{Suppb}(0)$, which is an obvious contradiction.

□

Remark 3.25. *In the context of error-correcting codes, Proposition 3.24 is equivalent with the well known fact that if $r < m$, then $R(r, m)$ consists of even codes only, see for instance [13].*

We end up by stating and proving three trivial propositions.

Proposition 3.26. $(1, \ldots, 1) \in V(f)$ *if and only if* $\mathrm{Suppb}(f)$ *is even.*

Proof. Suppose that $\mathrm{Suppb}(f)$ is even. Then it is immediate that $f(1, \ldots, 1) = 0$. On the other hand, if $\mathrm{Suppb}(f)$ is odd, then $f(1, \ldots, 1) = 1$. □

Proposition 3.27. $(0, \ldots, 0) \in V(f)$ *if and only if* $1 \notin \mathrm{Suppb}(f)$.

Proof. Suppose that $1 \notin \mathrm{Suppb}(f)$. Then $f(0, \ldots, 0) = 0$. But if $1 \in \mathrm{Suppb}(f)$, then $f(0, \ldots, 0) = 1$. □

Proposition 3.28. *If $f = f_1 \cdots f_m + 1$ as boolean polynomials and $|\mathrm{Suppb}(f_i)|$ is odd for $i = 1, \ldots, m$, then $f \neq 1$ and $(1, \ldots, 1) \notin V(f)$.*

Proof. It is enough to prove that, regarded as a boolean polynomial, the expression $f_1 \cdots f_m$ consists of an odd number of terms. If the product in $Z[x_1, \ldots, x_n]$, this is trivially true. But cancellations modulo $\mathbb{Z}_2$ and $x_i^2 + x_i$ are always done in pairs, hence $\mathrm{Suppb}(f_1 \cdots f_m)$ is odd. By Proposition 3.26, $(1, \ldots, 1) \notin V(f)$. □

4 Computational aspects

It is natural to ask whether the results developed so forth can be used in practice. Since the computational aspects is not the focus of this paper, we only focus on the question wether or not a boolean ideal $(x_1^2 + x_1, \ldots, x_n^2 + x_n, f_1, \ldots, f_m)$ has non-empty variety.

We have used the computer algebra program Macaulay2 [11] and the PolyBoRi [2] system to perform the computations and we compare the running time with the dire-running time for the corresponding Gröbner basis computation. The reason for choosing

Macaulay2 is twofold. On the one hand, we need a system which don't use a special representation of the boolean monomials and polynomials in order for the comparison to be fair. On the other hand, Macaulay2 has a built in package suited for performing computations within boolean ideals called BooleanGB. This package uses a bitwise representation of monomials. The reason for choosing PolyBoRi is because it is the fastest package available for computing Boolean Gröbner bases.

Our naive implementation consists only of calculating the product

$$(f_1 + 1) \cdots (f_m + 1)$$

and to check whether or not this expression equals zero (unsatisfiable/empty variety) or not (satisfiable/non-empty variety). In order for the comparison to be reasonable fair, we have not even used the trivial trick to compute the product degree by degree and stop as soon as we encounter a non-zero element.

The result is striking and shows at least that this direct method should be considered as a serious alternative to Gröbner basis calculations for boolean ideals, at least when it comes to testing satisfiability of the system.

Ex	#Var	#Gen	Sat	M2	M2 - Pack	M2 - Dir	PB	PB - Dir
1	13	3	Yes	204.739	33.986	0.003	6.66	0.00
2	13	3	Yes	495.16	304.98	0.01	10.12	< 0.01
3	15	4	Yes	> 1000	> 1000	0.01	81.76	< 0.01
4	19	4	Yes	> 1000	> 1000	0.01	> 1000	< 0.01
5	15	5	No	0.30	101.02	0.01	0.01	< 0.01
6	15	5	No	> 1000	< 0.01	0.11	0.01	< 0.01
7	34	17	Yes	< 0.01	< 0.01	> 1000	0.01	9.16
8	44	22	Yes	< 0.01	< 0.01	> 1000	0.01	> 1000

Table 1: Comparison between the direct approach and the approach using Gröbner bases in Macaulay2 version 1.4 and in PolyBoRi (within Sage 5.3). Timings are in seconds on a two year old MacBook. M2 denotes the method using the gb command in Macaulay2, M2 - Pack denotes the method using the booleanGB package and M2 - Dir denotes the direct method in Macaulay2 (without help from the package), PB denotes the Gröbner basis algorithm in PolyBoRi (called from Sage) and PB - Dir denotes the direct method in PolyBoRi. The package in M2 could not handle more than 32 variables.

The first and second example are truncated cyclic 13 -roots. The ideals are generated by $(x_1^2 + x_1, \ldots, x_{13} + x_1, f_1, f_2, f_3)$, where in the first example $f_1 = x_1 + x_2 + x_3 + x_4 + x_5 + x_6 + x_7 + x_8 + x_9 + x_{10} + x_{11} + x_{12} + x_{13}$, $f_2 = x_1x_2 + x_2x_3 + x_3x_4 + x_4x_5 + x_5x_6 + x_6x_7 + x_7x_8 + x_8x_9 + x_9x_{10} + x_{10}x_{11} + x_{11}x_{12} + x_{12}x_{13} + x_{13}x_1$ and $f_3 = x_1x_2x_3 + x_2x_3x_4 + x_3x_4x_5 + x_4x_5x_6 + x_5x_6x_7 + x_6x_7x_8$.

The second example is the same as the first except that here $f_3 = x_1x_2x_3 + x_2x_3x_4 + x_3x_4x_5 + x_4x_5x_6 + x_5x_6x_7 + x_6x_7x_8 + x_7x_8x_9 + x_8x_9x_{10} + x_9x_{10}x_{11} + x_{10}x_{11}x_{12} + x_{11}x_{12}x_{13} + x_{12}x_{13}x_1 + x_{13}x_1x_2$.

The generators (except the trivial ones) in the third example and fourth are random polynomials of degree two with 13 terms in each polynomial in 15 and 19 variables respectively.

The fifth and sixth example are modifications of the third. In the fifth we add the generators $x_1x_2 + 1$ and $(x + 1)(x_3 + 1)(x_4 + 1)(x_5 + 1) + 1$, which makes the system unsatisfiable. In the sixth we instead add the generator $x_1 \cdots x_{15} + 1$.

It is here very clear that the performance of the Buchberger algorithm depends upon choice. Selecting the pair $S(x_1x_2+1, x_1^2+x_1)$ in the fourth example gives x_1+1 after a trivial reduction, which is then used to reduce the second generator to 1, and we are done. This selection is obviously done early for the standard implementation, but late in the specialized package in Macaulay2.

In the same manner, computing the S-polynomial $S(x_i^2+x_i, x_1\cdots x_{15}+1)$ yields x_i+1. Repeating this for $i=1,\ldots,15$ gives the polynomials $x_1+1,\ldots,x_n+1$, so the only computation that is left to do is a fast reduction. In this case, the package outperforms the standard implementation.

The remaining two examples are constructed in a way which suites the Gröbner approach perfectly; it consists of the polynomials $(x_1x_2+x_1+x_2+1, x_3x_4+x_3+x_4+1,\ldots,x_{2n-1}x_{2n}+x_{2n-1}+x_{2n}+1)$. This is already, due to the Prime Criteria [5][2] and the fact that $S(x_ix_{i+1}+x_i+x_{i+1}+1, x_i^2+x_i) = x_i^2+x_i$, a Gröbner basis with respect to any term order. Here the problem of term expansion shows the weakness of the direct method. On the other hand, a more sophisticated implementation would detect that $x_1\cdots x_{2n} \in \mathrm{Suppb}((x_1x_2+x_1+x_2+1+1)\cdots(x_{2n-1}x_{2n}+x_{2n-1}+x_{2n}+1+1))$, and make use of Theorem 3.2 to detect that the system in satisfiable. (Using Proposition 3.24, we could even draw the stronger conclusion that the number of solutions is odd.)

Remark 4.1. *The question whether or not* $(f_1+1)\cdots(f_m+1)+1=0$ *as boolean polynomials is connected to the problem of Polynomial Identity Testing (PIT) [16]. The PIT problem is to determine whether or not a formula in* $\Bbbk[x_1,\ldots,x_n]$ *is zero, so the difference between this problem and our problem is that we are working modulo the ideal* $(x_1^2+x_1,\ldots,x_n^2+x_n)$.

5 Discussion

We have examined algebraic properties of ideals in $\mathbb{Z}_2[x_1,\ldots,x_n]$ containing the equations $x_1^2+x_1,\ldots,x_n^2+x_n$ by means of their defining polynomial. While one usually is in need of Gröbner bases to perform computations with respect to ideals in polynomials rings, we have shown that most computations can be done directly by using the defining polynomial.

There are many open questions related to the approach. First of all, the connection between the defining polynomial and the corresponding variety should be carefully studied. Here, the work by Harrison from the sixties [12], which classifies boolean polynomials by the general linear and affine groups, might be of interest.

The computational strength of the direct approach is that is a *direct* way of solving the SAT problem and that *no extra information* about the problem is given as a side effect. Compare this with the DPLL-based methods where a solution is actually returned in case the system is satisfiable, or with the Gröbner based method, where one, a side effect, gets information about the size of the variety.

The drawback is of course term expansion. Hopefully, term expansion can hopefully be avoided by making use of PIT-related algorithms. One might also ask whether it is possible to use branching techniques like in the DPLL influenced algorithms.

[2]The criteria is called "The first criteria" in [5], but the term "Prime Criteria" is folklore nowadays.

References

[1] M. Brickenstein, Boolean Gröbner bases. Theory, algorithms and applications Berlin: Logos Verlag; Kaiserslautern: TU Kaiserslautern, Fachbereich Mathematik (Diss. 2010) (ISBN 978-3-8325-2597-2/pbk). x, 149 p. (2010).

[2] M. Brickenstein, A. Dreyer, PolyBoRi: A framework for Gröbner-basis computations with Boolean polynomials. J. Symb. Comput. 44 (2009), no. 9, 1326 – 1345.

[3] M. Brickenstein, A. Dreyer, Gröbner-free normal forms for Boolean polynomials, J. Symb. Comput. 48, no. 1, 2013, 37–53.

[4] M. Brickenstein, A. Dreyer, G.-M. Greuel, F. Seelisch, O. Wienand, New developments in the theory of Gröbner bases and applications to formal verification, J. Pure Appl. Alg. 213 (2009), no. 8, 1612–1635.

[5] B. Buchberger, A criterion for detecting unnecessary reductions in the construction of Groebner bases. Proc. EUROSAM 79, LNCS 72 (1979), 3–21.

[6] B. Buchberger, M. Möller. The construction of multivariate polynomials with preassigned zeroes, Computer Algebra: EUROCAM .82 (J. Calmet, ed.), Lecture Notes in Computer Science 144 (1982), 2431.

[7] L. Cerlienco, M. Mureddu. From algebraic sets to monomial linear bases by means of combinatorial algorithms. Discrete Math. 139 (1995), 7387.

[8] C. Condrat, P. Kalla, A Gröbner basis approach to CNF-formulae preprocessing. Proceeding TACAS'07 Proceedings of the 13th international conference on Tools and algorithms for the construction and analysis of systems (2007), 618-631.

[9] M. Davis, G. Logemann, D. Loveland, A Machine Program for Theorem Proving Communications of the ACM 5 (1962), 394397. doi:10.1145/368273.368557.

[10] V. Gerdt, M. Zinin, A pommaret division algorithm for computing Grobner bases in boolean rings, ISSAC '08 (2008), 95-102.

[11] D. R. Grayson, M.E. Stillman, Macaulay2, a software system for research in algebraic geometry, Available at http://www.math.uiuc.edu/Macaulay2/

[12] M. A. Harrison, On the classification of Boolean functions by the general linear and affine groups. J. Soc. Indust. Appl. Math. 12 (1964), 285299.

[13] J. Hendricus V. Lint, Introduction to Coding Theory. Springer-Verlag New York, 1982. ISBN: 0387112847.

[14] S. Lundqvist, Vector space bases associated to vanishing ideals of points, J. Pure Appl. Alg. 214 (2010), no. 4, 309–321.

[15] Y. Sato, S, Inoue, A. Suzuki, K. Nabeshima, K. Sakai, Boolean Gröbner bases. J. Symb. Comput. 46 (2011), no. 5, 622–632.

[16] J.T. Schwartz, Fast Probabilistic Algorithms for Verification of Polynomial Identities. Journal of the ACM 27 (2011), no. 4, 701 – 717.

[17] W. A. Stein et al., Sage Mathematics Software (Version 5.3), The Sage Development Team, 2012, http://www.sagemath.org.

[18] C. Zengler, W. Küchlin, Communications in Computer Algebra archive 45 (2011), no. 1/2, 141–142.